МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образование «Белорусский государственный технологический университет»

**«**ИССЛЕДОВАНИЕ СТЕГАНОГРАФИЧЕСКОГО МЕТОДА НА ОСНОВЕ ПРЕОБРАЗОВАНИЯ НАИМЕНЕЕ ЗНАЧАЩИХ БИТОВ**»**

Студент: Бутурля Р.А.

ФИТ 3 курс 5 группа

Вариант 3

Преподаватель: Савельева М. Г.

Минск 2023

1. Разработать собственное приложение, в котором должен быть реализован метод НЗБ.

Для записи данных в файл по горизонтали был разработан метод HideText. Сообщение записывается в младшие биты изображения. Листинг метода представлен ниже.

|  |
| --- |
| public static Bitmap HideText(string text, Bitmap bmp)  {  State state = State.Hiding;  int charIndex = 0;  int charValue = 0;  long pixelElementIndex = 0;  int zeros = 0;  int R = 0, G = 0, B = 0;  for (int i = 0; i < bmp.Height; i++)  {  for (int j = 0; j < bmp.Width; j++)  {  Color pixel = bmp.GetPixel(j, i);  R = pixel.R - pixel.R % 2;  G = pixel.G - pixel.G % 2;  B = pixel.B - pixel.B % 2;  for (int n = 0; n < 3; n++)  {  if (pixelElementIndex % 8 == 0)  {  if (state == State.FillingWithZeros && zeros == 8)  {  if ((pixelElementIndex - 1) % 3 < 2)  {  bmp.SetPixel(j, i, Color.FromArgb(R, G, B));  }  return bmp;  }  if (charIndex >= text.Length)  {  state = State.FillingWithZeros;  }  else  {  charValue = text[charIndex++];  }  }  switch (pixelElementIndex % 3)  {  case 0:  {  if (state == State.Hiding)  {  R += charValue % 2;  charValue /= 2;  }  }  break;  case 1:  {  if (state == State.Hiding)  {  G += charValue % 2;  charValue /= 2;  }  }  break;  case 2:  {  if (state == State.Hiding)  {  B += charValue % 2;  charValue /= 2;  }  bmp.SetPixel(j, i, Color.FromArgb(R, G, B));  }  break;  }  pixelElementIndex++;  if (state == State.FillingWithZeros)  {  zeros++;  }  }  }  }  return bmp;  } |

Листинг 1 – Метод HideText

Для записи данных в файл по вертикали был разработан метод HideTextVert. Метод аналогичен предыдущему.

|  |
| --- |
| public static Bitmap HideTextVert(string text, Bitmap bmp)  {  State state = State.Hiding;  int charIndex = 0;  int charValue = 0;  long pixelElementIndex = 0;  int zeros = 0;  int R = 0, G = 0, B = 0;  for (int i = 0; i < bmp.Width; i++)  {  for (int j = 0; j < bmp.Height; j++)  {  Color pixel = bmp.GetPixel(j, i);  R = pixel.R - pixel.R % 2;  G = pixel.G - pixel.G % 2;  B = pixel.B - pixel.B % 2;  for (int n = 0; n < 3; n++)  {  if (pixelElementIndex % 8 == 0)  {  if (state == State.FillingWithZeros && zeros == 8)  {  if ((pixelElementIndex - 1) % 3 < 2)  {  bmp.SetPixel(j, i, Color.FromArgb(R, G, B));  }  return bmp;  }  if (charIndex >= text.Length)  {  state = State.FillingWithZeros;  }  else  {  charValue = text[charIndex++];  }  }  switch (pixelElementIndex % 3)  {  case 0:  {  if (state == State.Hiding)  {  R += charValue % 2;  charValue /= 2;  }  }  break;  case 1:  {  if (state == State.Hiding)  {  G += charValue % 2;  charValue /= 2;  }  }  break;  case 2:  {  if (state == State.Hiding)  {  B += charValue % 2;  charValue /= 2;  }  bmp.SetPixel(j, i, Color.FromArgb(R, G, B));  }  break;  }  pixelElementIndex++;  if (state == State.FillingWithZeros)  {  zeros++;  }  }  }  }  return bmp;  } |

Листинг 2 – Метод HideTextVert

Для получения данных из файла были разработаны методы ExtractText и ExtractTextVert. Методы преобразуют файл в битовый массив и считывают данные из него. Методы служат для горизонтального и вертикального считывания соответственно.

|  |
| --- |
| public static string ExtractText(Bitmap bmp)  {  int colorUnitIndex = 0;  int charValue = 0;  string extractedText = "";  for (int i = 0; i < bmp.Height; i++)  {  for (int j = 0; j < bmp.Width; j++)  {  Color pixel = bmp.GetPixel(j, i);  for (int n = 0; n < 3; n++)  {  switch (colorUnitIndex % 3)  {  case 0:  {  charValue = charValue \* 2 + pixel.R % 2;  }  break;  case 1:  {  charValue = charValue \* 2 + pixel.G % 2;  }  break;  case 2:  {  charValue = charValue \* 2 + pixel.B % 2;  }  break;  }  colorUnitIndex++;  if (colorUnitIndex % 8 == 0)  {  charValue = ReverseBits(charValue);  if (charValue == 0)  {  return extractedText;  }  char c = (char)charValue;  extractedText += c.ToString();  }  }  }  }  return extractedText;  } |

Листинг 3 – Метод ExtractText

|  |
| --- |
| public static string ExtractTextVert(Bitmap bmp)  {  int colorUnitIndex = 0;  int charValue = 0;  string extractedText = "";  for (int i = 0; i < bmp.Width; i++)  {  for (int j = 0; j < bmp.Height; j++)  {  Color pixel = bmp.GetPixel(j, i);  for (int n = 0; n < 3; n++)  {  switch (colorUnitIndex % 3)  {  case 0:  {  charValue = charValue \* 2 + pixel.R % 2;  }  break;  case 1:  {  charValue = charValue \* 2 + pixel.G % 2;  }  break;  case 2:  {  charValue = charValue \* 2 + pixel.B % 2;  }  break;  }  colorUnitIndex++;  if (colorUnitIndex % 8 == 0)  {  charValue = ReverseBits(charValue);  if (charValue == 0)  {  return extractedText;  }  char c = (char)charValue;  extractedText += c.ToString();  }  }  }  }  return extractedText;  } |

Листинг 4 – Метод ExtractTextVert

Изображение до и после внедрения туда данных:



Рисунок 1 – Исходное изображение



Рисунок 2 – Изображение с данными

Как мы видим на рисунках 1 и 2 изображение практически не изменилос.

Зашифрованные данные могут размещаться как вертикально, так и горизонтально. В зависимости от направлнения запись в файл данных будет регулироваться специальной конструкцией. Пример ниже.

|  |
| --- |
| for (int i = 0; i < bmp.Width; i++)  {  for (int j = 0; j < bmp.Height; j++)  {  Color pixel = bmp.GetPixel(j, i);  for (int n = 0; n < 3; n++)  {  switch (colorUnitIndex % 3)  {  case 0:  {  charValue = charValue \* 2 + pixel.R % 2;  }  break;  case 1:  {  charValue = charValue \* 2 + pixel.G % 2;  }  break;  case 2:  {  charValue = charValue \* 2 + pixel.B % 2;  }  break;  }  colorUnitIndex++;  if (colorUnitIndex % 8 == 0)  {  charValue = ReverseBits(charValue);  if (charValue == 0)  {  return extractedText  }  char c = (char)charValue;  extractedText += c.ToString();  }  }  }  } |

Листинг 4 – Метод вставки по вертикали

Цветовая матрица изображения представлена ниже.



Рисунок 3 – Цветовая матрица до шифрования



Рисунок 4 – Цветовая матрица после шифровании

**Вывод**

Изучил стеганографический метод встраивания\* /извлечения тайной информации с использованием электронного файла-контейнера на основе преобразования наименее значащих битов (НЗБ), приобрел практические навыки программной реализации данного метода.